

Scientific Best-Practices for Recurring Problems in Computer Security R & D

Daniel Bilar
Director of Research, Siege Technologies
Manchester, New Hampshire, USA
dbilar@acm.org
@daniel_bilar

SyScan 2014
Singapore
April 3rd, 2014

Outline of Talk

- Dynamic SyScan update: Car CAN R & D
- Signals/ Side Channels
- Side Channels & Methods
 - Detection
 - Representation
 - Analysis
- Case studies
 - ROPE, Bochspwn, Cyber Mission Planning
- Epilogue/Blueprint

Thanks for help/
inspiration and
appreciation to

Thomas **Dullien**
Ero **Carrera**
Alex **Sotirov**
Travis **Goodspeed**
Anna **Shubina**
Sergey **Bratus**
Rebecca **Shapiro**
Jason **Geffner**
Jon **Stuart**
Georg **Wicherski**
Mateusz **Jurczyk**
Gynvael **Coldwind**

+ many **more**

Addendum: Car CAN bus hacking

- Did a subset of what Chris and Charlie did in March 2013, presented at RECon June 2013
- **“Hot-Wiring of the Future”**
 - Lots of tips, (free, \$) tools, workflow/methodology, Costs: \$6k (2 cars)
- Sponsored 3 undergraduate students (knew nothing at all about RE) who learned how to reverse, hook up boards, use goodThopter and denial of view/manipulation of CAN dashboard in 3 months

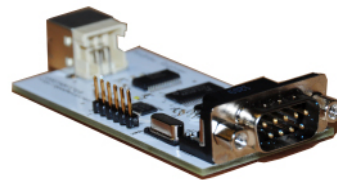
<http://tinyurl.com/CarCAN2013>



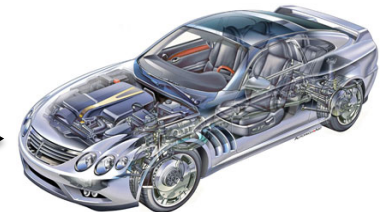
User



Software Package

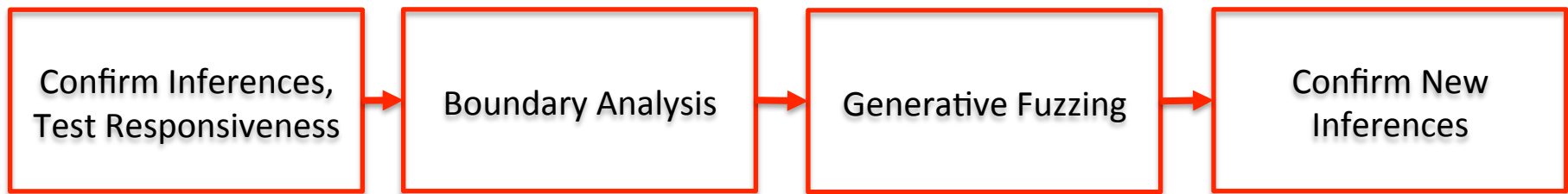


GoodThopter10



CAN Bus

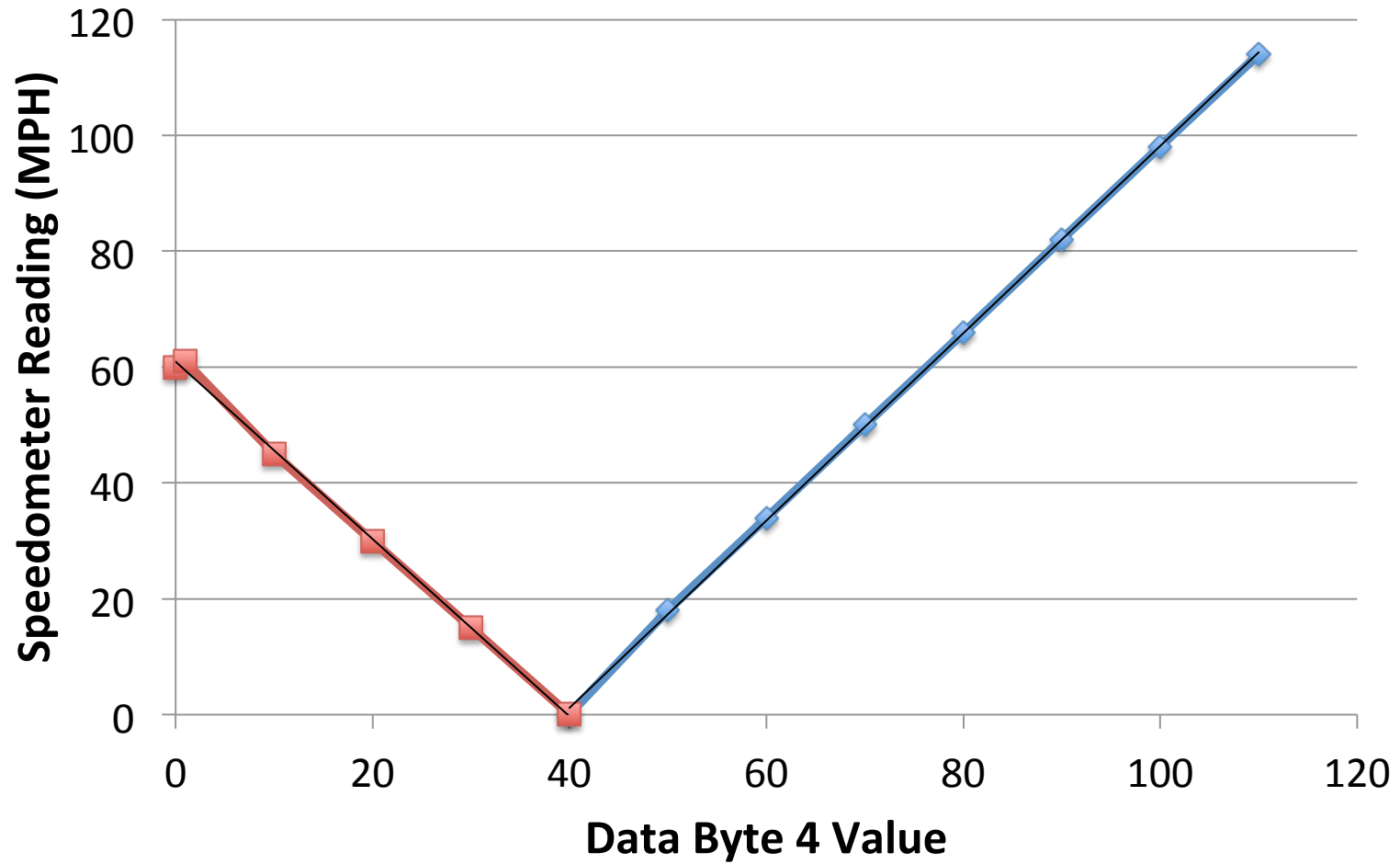
Work Accomplished - Methodology



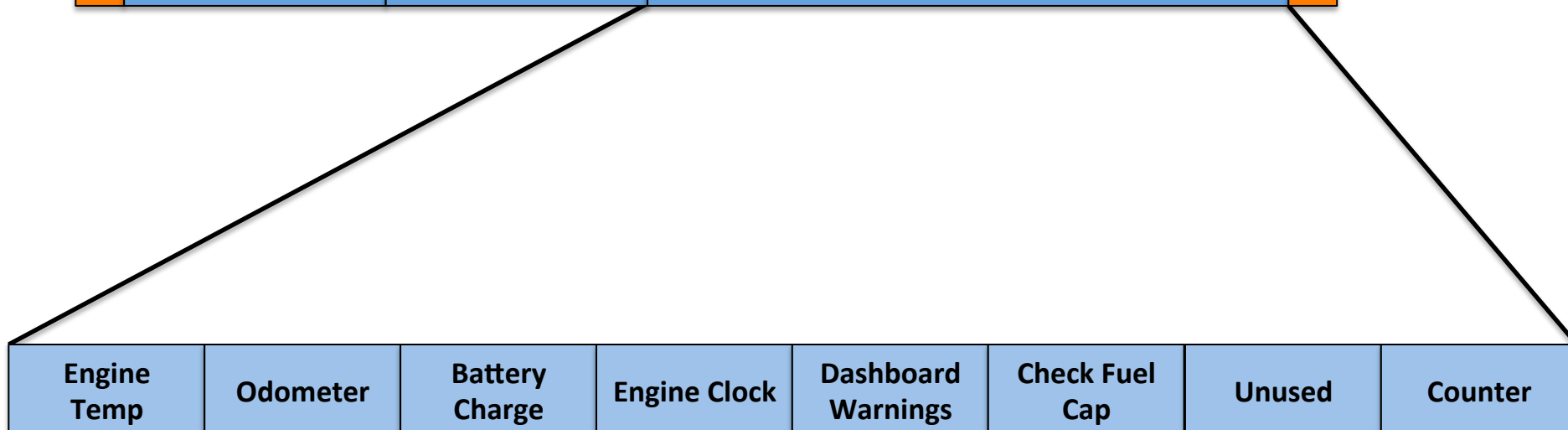
A Case Study – ID 513

ArbID	Fuzzing Response	Refined Inferences
513	Dashboard components change	Bytes 0, 1 = RPM
		Byte 4 = Speedometer

Higher Level Protocol: ID 513



Higher Level Protocol: ID 1056

1056**DLC = 8**

Signals: Side Channel

- Side channels = observables that are emitted by active systems
- On a computer system, these can be observed time, power, OS events, EM radiation, characteristic acoustic spectral signature and more
- 2014: Mark Stoettinger's NTU group is doing cutting edge punctuated hardware magnetic field side channel analysis (SCA)

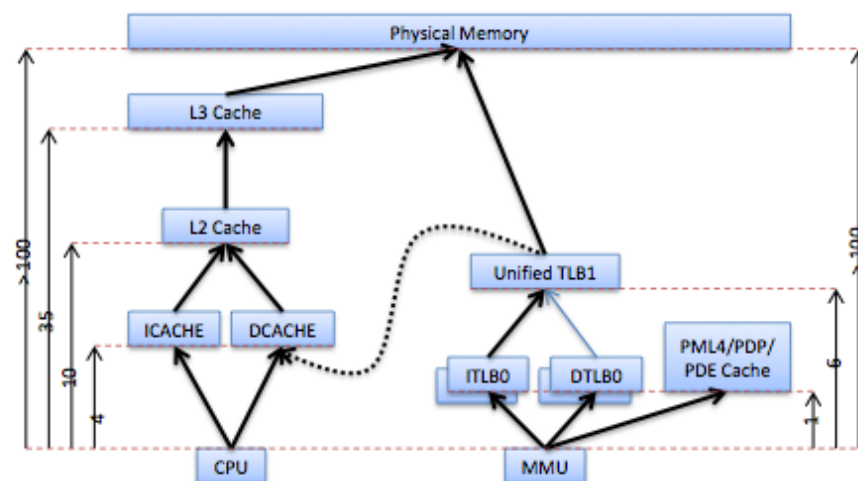


Figure 2. Intel i7 memory hierarchy plus clock latency for the relevant stages

Graph from [Hund2013]. Generic timing side channel attack against MMU system to infer information about the privileged address space layout

Some innovative attacks: data structures (2007 timing attacks against databases), protocols and underlying algorithms (2007 QoS attacks against balancing algorithms, MMU (cache) and more

Side Channels as Consilient Evidence

- Generalize: Use side channel evidence to reason about system/sub-system (internal states)
- Whewell's "Consilience of Induction"
 - Concept of aggregate evidence
 - Convergence of several, ideally independent hypotheses serves to strengthen conclusion
- Question: What side channels are available, easy to access, analyze, expressive, type I/II error etc

looks like a duck
quacks like a duck
has duck genome
hangs out with ducks
Pope tells you it is a duck
lives in water
does ducky things
"I am your father"



Duck Vader ☺

Three Questions on Signals

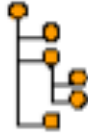
- Want actionable handling of higher dimensional signal dynamics as they occur in live computer systems as side channels
 - Actionable is to be understood as useful in practice
 - Higher dimensional refers to six or more data dimensions
 - Dynamics emphasizes the signal's unfolding in temporal and feature space
- We'll discuss
 - How to represent such signals
 - How to detect signals with various characteristics
 - How to prevent or mitigate the leaking of such signals

Signal Representation: Visuals

Property	Elaboration
Well-boundedness	Do changes in the representation express changes in the data, and not particular algorithmic artifacts?
Actionability	Does the representation support formulating hypotheses and subsequently testing them?
Specificity	Does the representation fit the nature of the domain whence the data sprung? Is it amenable to revealing the essence of the underlying process?
Parsimony	Does the representation uses the absolute minimum number of attributes necessary for specificity?
High-Dimensionality	Does the data representation support enough dimensions to facilitate comprehension of the underlying process ?
Scalability	Can large orders-of-magnitude rescaling with concomitant drill down/zoom out be intelligibly accommodated?
Perspective	Can dependent and independent dimensions be changed to enable different process view of the data ?

Table 1: Representation Evaluation Criteria

Signal Representation: Visuals & more

Relationship Type:	General Similarity	Explicit Reference	Field/Value Co-occurrence	Parent/Child	Spatial	Temporal
Model Type:	<i>Vector-space</i> 	<i>Network</i> 	<i>Multidimensional Index</i> 	<i>Hierarchical</i> 	<i>Spatial</i> 	<i>Ordinal Index</i> 
Examples:	<i>Reports, articles, DB records</i>	<i>References & citations, hyperlinks</i>	<i>DB records, document metadata</i>	<i>File paths, taxonomies, IP addresses</i>	<i>Geolocations, CAD models</i>	<i>Event descriptions</i>

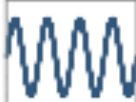
- Fundamental cognitive limits for visuals very hard to overcome for high dimensional representation
 - See Starlight project, NIDS graphs, Edward Tufte book series, Marty “Applied Security Visualization”, Paley “Visual Analytics”
- Rich gamut of of human senses remain neglected
- Aural (hearing), haptic (touch), vestibular (balance and acceleration), kinesthetic, thermoception (temperature), etc

Signal Detection: MINE Statistics

Maximal Information-based
Nonparametric Exploration
(MINE) statistics

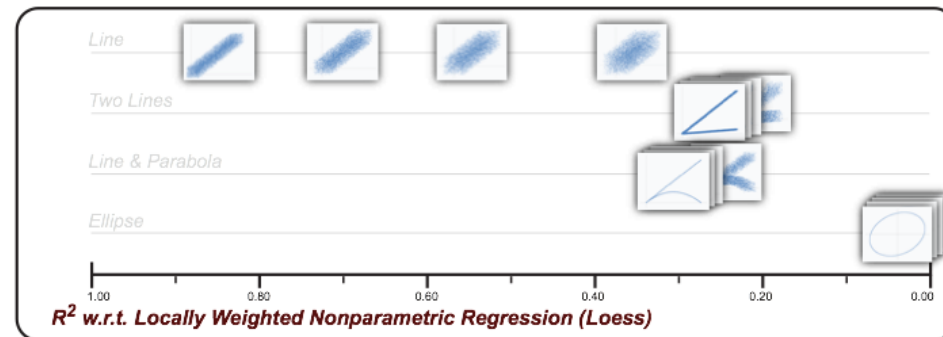
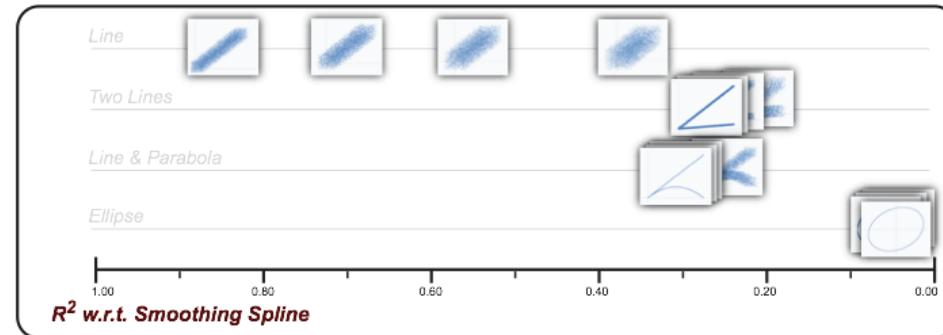
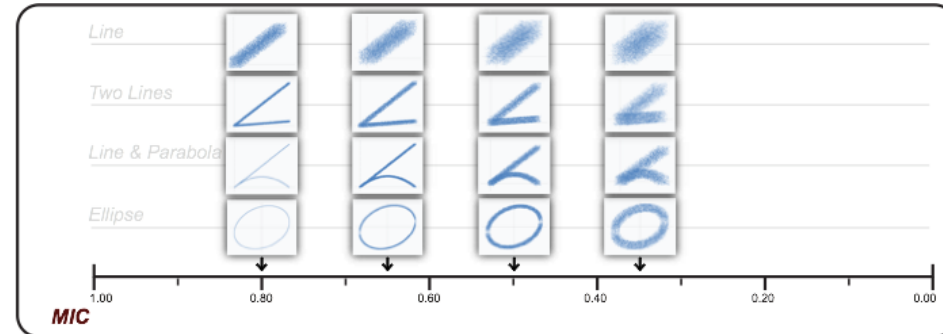
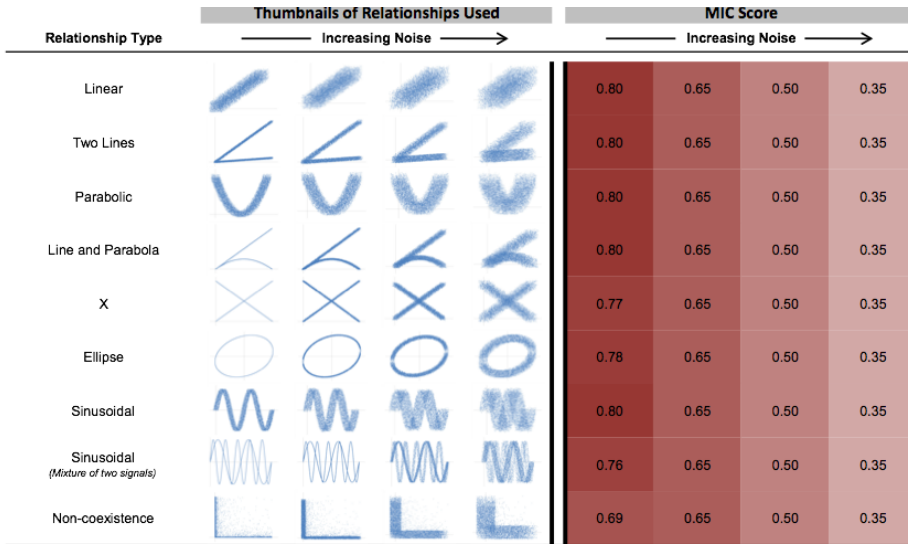
- General: **Captures wide range of associations between pairs of variables** (linear, exponential, periodic, non-functions)
- Equitable: Assigns similar scores to equally noisy relationships of different types [Reshef2011sup]

Table from [Reshef2011]

Data	MIC	MAS	MEV	MCN
	1.00	0.00	1.00	2.00
	1.00	0.74	1.00	3.00
	1.00	0.89	1.00	4.00
	1.00	0.69	1.00	2.56
	0.79	0.16	0.70	6.91
	0.71	0.03	0.32	6.87
	0.46	0.19	0.22	6.98

MIC captures **general relationship strength**
MIC-r² captures **non-linearity** (Not shown)
MCN captures **complexity**
MAS captures departure from **monotonicity**
MEV captures closeness to being a **function**

Signal Detection: Association 'Types'



Signal Prevention: Side Channel Leaks

- **Possible to control rate but not eliminate side channels**
- Recently, Goldwasser (MIT/Technion) and Rothblum offer a practical way forward
- Resisting leakage at *design time* and offers progress towards *formulation of automatic approaches* that generate “leakage-resilience” programs for a wide range of side channel attacks
- Proved that for any computationally unbounded A observing the results of computationally unbounded leakage functions, will learn no more from its observations than it could given blackbox access only to the input-output behavior of P
- Result is *unconditional* and does not rely on any secure hardware components

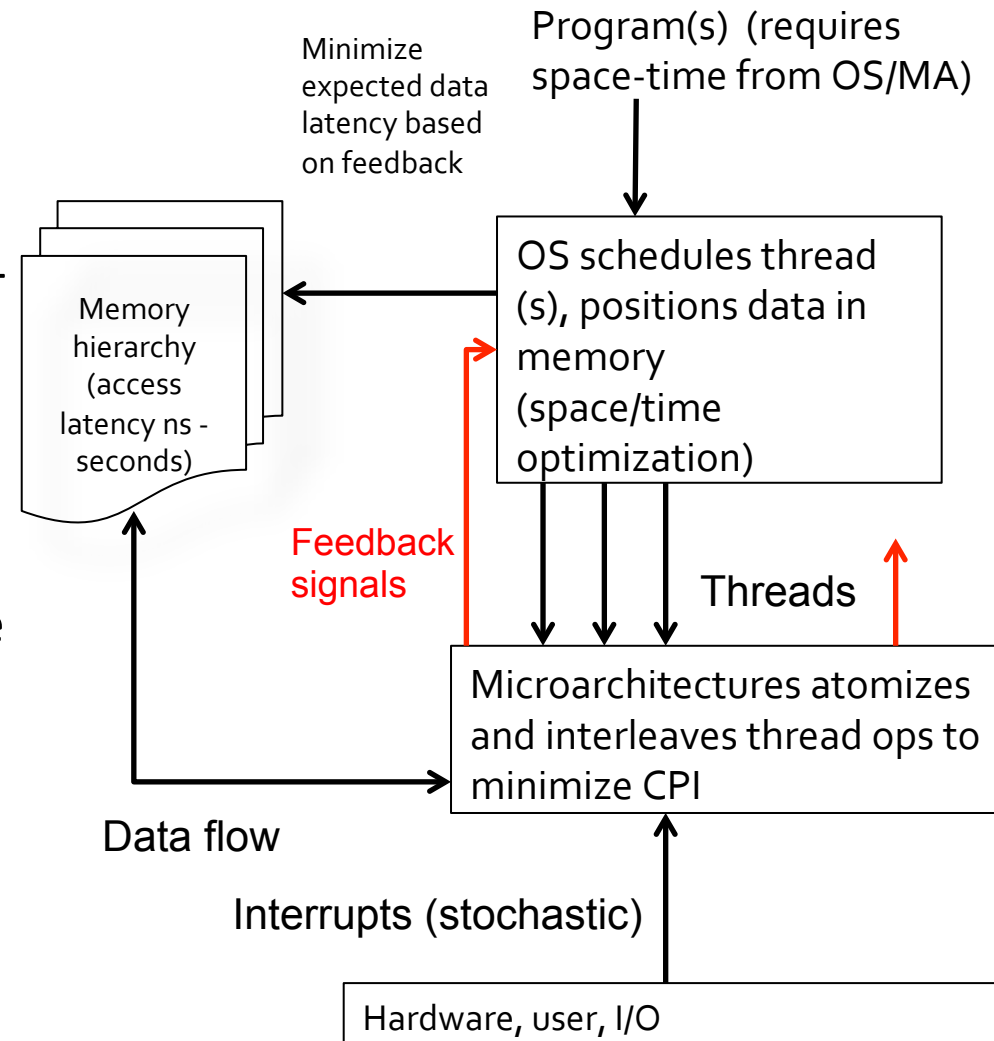
Cyber Mission Planning

- Cyber-operations have potential to be more pinpointed than kinetic counterpart
 - Minimize collateral damage by ‘crisp’ targeted operations
- However, unlike kinetic planning (centuries of well-understood natural laws), **cyber-planning lacks foundational corpus of predictive laws**

“Natural Laws for/of/in Virtual Reality”

Thread / Process/ Cache Execution

- Behavior arises as a complex interaction of timing of memory requests (program behavior), cache coherence protocol (dependent on MA), or thread pre-emption (depending on OS) [Alistarh2014]
- Modern operating systems (OS) and microarchitectures (MA) = dynamic complex feedback system that tries to continuously minimize CPI (cycles per instruction)
 - Memory latency is bottleneck, hence memory hierarchies from ns to s
- **OS and MA continuously solve a time-space optimization problem to 'flatten' (parallelize) sequential processing**



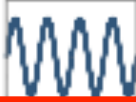
Signal Detection: MINE/MIC

Maximal Information-based
Nonparametric Exploration (MINE)
statistics

Intuition: (Simple) asset 'signals' are
reflected in convex/concave
parabola-type curves in time

Identify signals that are less periodic
(lower MAS), less linear (MIC-r²),
but still a function (higher MEV)

*“not a heartbeat, not a shooting star,
but still a function”*

Data	MIC	MAS	MEV	MCN
	1.00	0.00	1.00	2.00
	1.00	0.74	1.00	3.00
	1.00	0.89	1.00	4.00
	1.00	0.69	1.00	2.56
	0.79	0.16	0.70	6.91
	0.71	0.03	0.32	6.87
	0.46	0.19	0.22	6.98

MIC captures **general relationship strength**
MIC-r² captures **non-linearity** (Not shown)
MCN captures **complexity**
MAS captures departure from **monotonicity**
MEV captures closeness to being a **function**

Issue: Experimental Factors

- Free NIST tools: Automated Combinatorial Testing for Software (ACTS) and Coverage Measurement (CCM)
- Cut down combinatorial explosion: 2-way, 3-way, n-way testing

Asset-Target Matching

- Say asset tested on configuration A and it has 18 categories (e.g. language, OS/patch, service running, workload, etc) @ dozen of values
- How similar is the unknown configuration B to A?
- Question of **distance**
 - Easy enough for ratio data (like Kelvin), much harder for categorical data (like OS type)

- Table shows 14 categorical distance (similarity) measures
- Differ primarily how they weigh matches and mismatches between categories

Table from [Boriah2013]

Measure	n_k	$\{f_k(X_k), f_k(Y_k)\}$	
		$X_k = Y_k$	$X_k \neq Y_k$
<i>Overlap</i>	$\propto n_k^2$	1	0
<i>Eskin</i>		1	0
<i>IOF</i>		1	$\propto 1/(\log f_k(X_k) \log f_k(Y_k))$
<i>OF</i>		1	$\propto \log f_k(X_k) \log f_k(Y_k)$
<i>Lin</i>		$\propto 1/\log f_k(X_k)$	$\propto 1/\log(f_k(X_k) + f_k(Y_k))$
<i>Lin1</i>		$\propto 1/\log f_k(X_k)$	$\propto 1/\log f_k(X_k) - f_k(Y_k) $
<i>Goodall1</i>		$\propto (1 - f_k^2(X_k))$	0
<i>Goodall2</i>		$\propto f_k^2(X_k)$	0
<i>Goodall3</i>		$\propto (1 - f_k^2(X_k))$	0
<i>Goodall4</i>		$\propto f_k^2(X_k)$	0
<i>Smirnov</i>		$\propto 1/f_k(X_k)$	$\propto 1/(f_k(X_k) + f_k(Y_k))$
<i>Gambaryan</i>		Maximum at $f_k(X_k) = \frac{N}{2}$	0
<i>Burnaby</i>	$\propto 1/n_k$	1	$\propto 1/\log f_k(X_k), \propto 1/\log f_k(Y_k)$
<i>Anderberg</i>		$\propto 1/f_k^2(X_k)$	$\propto f_k(X_k)f_k(Y_k)$

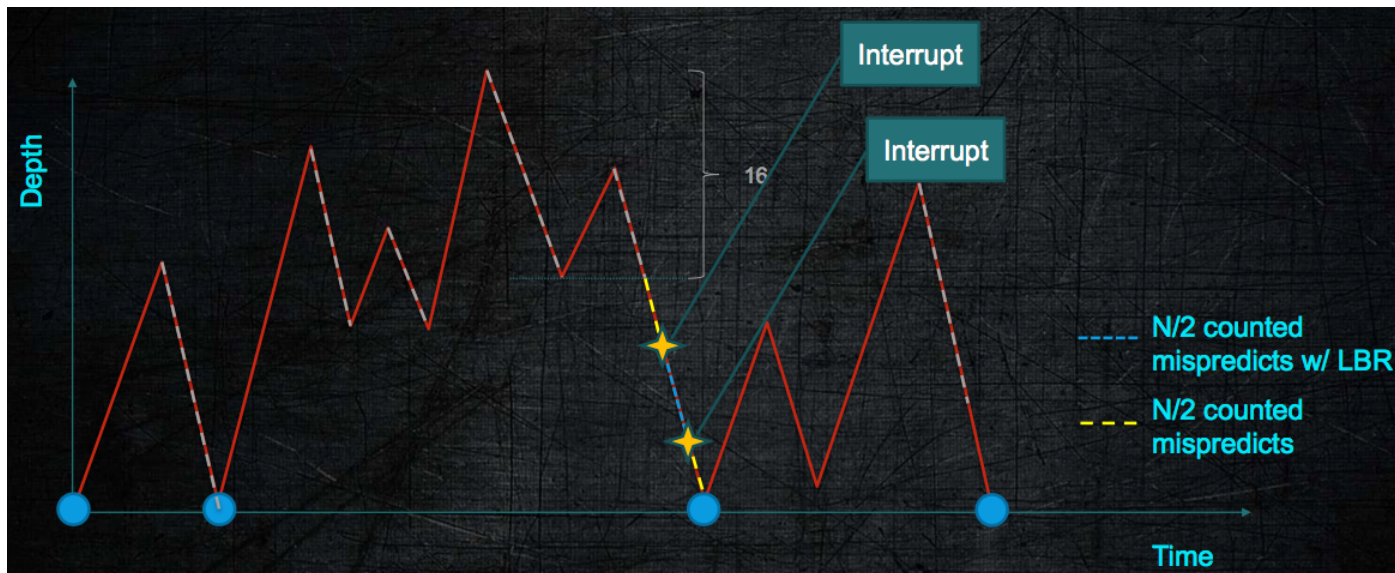
Table 3: Relation between per-attribute similarity, $S(X_k, Y_k)$ and $\{n_k, f_k(X_k), f_k(Y_k)\}$.

- ‘Success’ functions are not smooth but stepped
- Can shove 10cm³ (~elastic) object through 9.8cm³ hole –> smooth success degradation
- Difference between patch1 and patch2 is difference between works/doesn’t work -> step/catastrophic success degradation

Case study: ROPe

- ROPe: Detection of kernel-level ROP through branch return mispredictions
- 16 (N) entry shadow stack of call- sites / return addresses
 - 0x89 – BR_MISP_EXEC.*: mispredicted executed branches
 - 0x800 – .RETURN_NEAR: normal, near ret
 - 0x8000 – .TAKEN: unconditional branch
- PMC interrupts after certain number of mispredictions ($N/2 = 8$)
 - Upon interrupt, handler checks MSR Last Branch Recording (LBR) whether targets of the previously executed instructions are preceded by an instruction
 - If not -> likely ROP (chain) induced

Graph from G. Wicherski, SysCan 2013



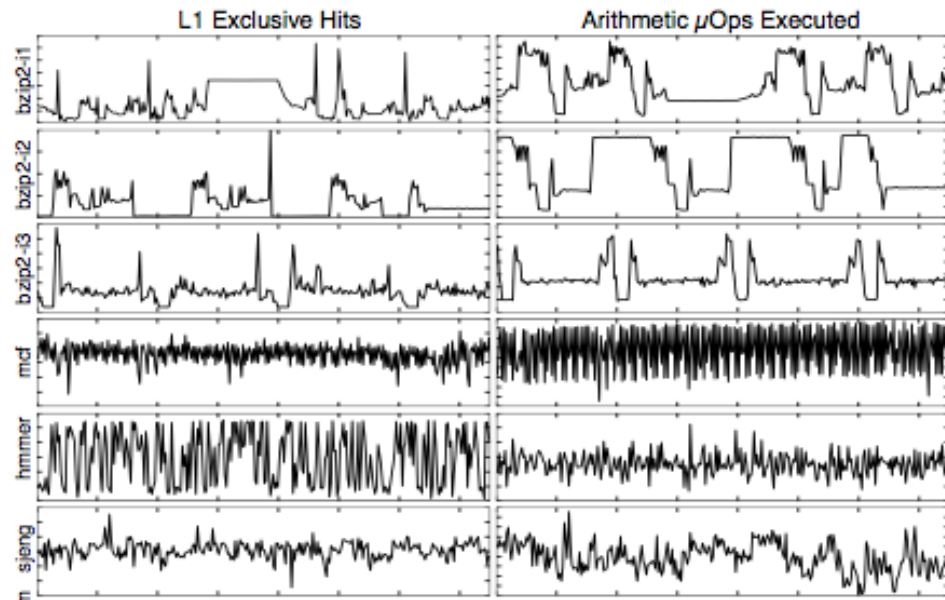
Started telling Travis Goodspeed at RECon 2013 about this and after less than 8 seconds he exclaims and I quote: *Holy cow! That's a \$^&* brilliant idea once you understand it!* 😊

Suggestions for ROPe

- Generalize ROP/0x8889 insight : Side channel ‘spectral signature’ for variety of interesting attacks
 - JOP, ‘weird machine’-inducers, hardware-based attacks, ...
 - Additional OS/MA vents

Workplan (high-level):

- Identify signals of interest
 - Scope with MINE [Reshef2011]
- Signal periodicity analysis
 - DSP, System Identification tools
- Scientifically valid experiment setup
 - Use procedures [Mont2012]

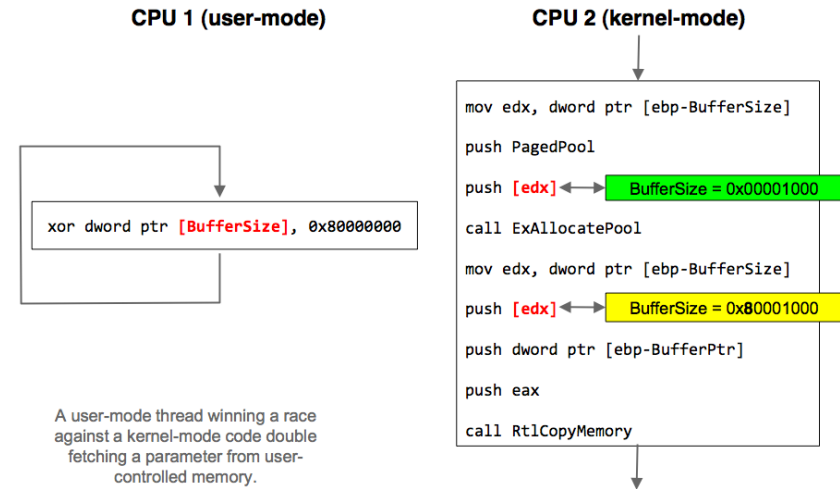


PMC measurements over time for programs in SPEC benchmark suite. Graph from [Demme2013]. Good results with 4-dim spectral signature from x86 MA events

- 0x0440 -- L1D_CACHE_LD.E_STATE
- 0x0324 -- L2_RQSTS.LOADS
- 0x03b1 -- UOPS_EXECUTED.PORT (1 or 2)
- 0x7f88 -- BR_INST_EXEC.ANY

Case study: Bowspwn

- Study of Double fetch operations
 - Two virtual address reads from kernel mode thread close in time
 - Virtual address concurrently writable by ring-3 threads
- Assumption of value consistency over time gives rise to race condition
 - User address space is shared across ring0 / ring3
 - User-mode memory regions can be modified at any time by concurrent ring3 thread
- Bochspxn idea: Extend time window between value Check and value Use to give ring3 attack thread opportunity to modify value
 - How? Bleed time by slow cache line and page boundaries, non-cacheability, TLB flushing (2500x slowdown achieved)
- 69 (!) LaTeX pages at SyScan 2013
 - Fundamental applied security paper, vital for safer concurrent programming [JC2013a]
 - Refinements: Flip interval dependence on value (binary, arithmetic) types, logistic S-curve discussion [JC2013b]



Graphs from [JS2013a]

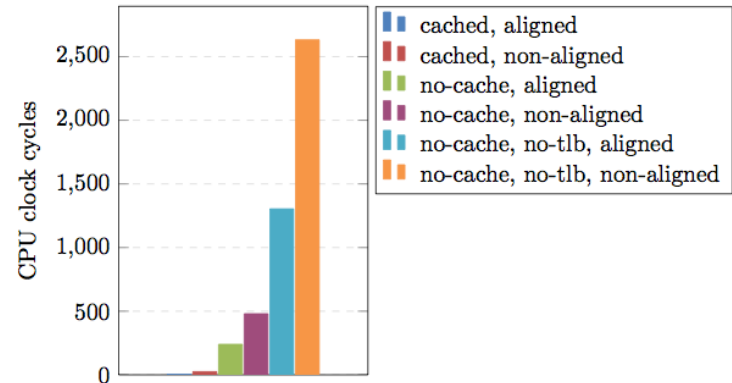


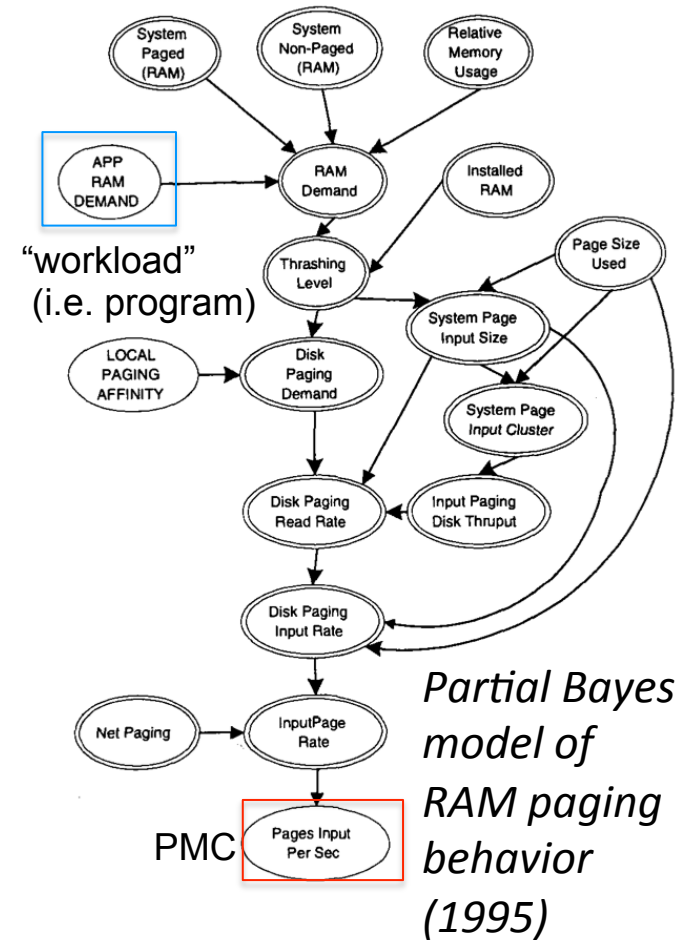
Figure 9: Cycle costs of 32-bit memory references with different attributes applied to the memory region and its address.

Suggestions for Bochs-pwn

	Embed	SPEC	DB	Games	ML	CAD	HPC
1. Finite State Mach.							
2. Circuits							
3. Graph Algorithms							
4. Structured Grid							
5. Dense Matrix							
6. Sparse Matrix							
7. Spectral (FFT)							
8. Dynamic Prog							
9. Particle Methods							
10. Backtrack/B&B							
11. Graphical Models							
12. Unstructured Grid							

12 high level computation language patterns mined from seven general application areas (green & blue rare) [Asan2009]

Practical: Find low level assembly pattern translation and investigate susceptibility to double fetch and resulting 'distortions'/ error



Theory: Investigate multi-core control systems (system scheduler, Paging) and 'bring out' assumptions

Epilogue: Methods ‘Blueprint’

- Signal Selection
 - Construct model of system
 - Identify side channel observables (OS/MA events & others)
 - Scope SCO 's MINE properties
 - Use MIC/MINE statistics [Reshef2011]
- Signal Representation & Analysis
 - Octave (free but not powerful enough), MATLAB (best choice)
 - Boxplots, Probability Plots
 - Toolboxes: Statistics, DSP, System Identification
 - Machine Learning
 - Internalize [Dom2012]
 - Select ML procedures from [Murph2012] *appropriate for and deduced from a system model and the signals' macro-properties*
 - [PMTK3] for Bayesian reasoning/modeling
- Scientific Experiments [Mont2012]
- Specialized tools:
 - Signal Analysis: Eureka [Lipson2009]
 - Signal Representation: Viewpoints [Gazis2010]

Pro-Tip ML

Likelihood	Prior	Name	Section
Gaussian	Uniform	Least squares	7.3
Gaussian	Gaussian	Ridge	7.5
Gaussian	Laplace	Lasso	13.3
Laplace	Uniform	Robust regression	7.4
Student	Uniform	Robust regression	Exercise 11.12

Table 7.1 Summary of various likelihoods and priors used for linear regression. The likelihood refers to the distributional form of $p(y|\mathbf{x}, \mathbf{w}, \sigma^2)$, and the prior refers to the distributional form of $p(\mathbf{w})$. MAP estimation with a uniform distribution corresponds to MLE.

- Know what features your favourite ML algo selects and weighs
- Many blind spots possible
- Study Domingos (2013)!

Current/upcoming R & D topics

- Added to the talk is a short WP with a selection of R & D issues
 - Concurrency Attacks
 - Compositional Security
 - Systemic Computer Security
- All these in my humble opinion benefit from SCA

Concurrency Attacks

- Even though we increasingly rely on concurrent execution, such programs are much more difficult to write, test, debug.
 - Potential for serious *concurrency errors* in many widespread concurrent programs, enabling feasible *concurrency attacks*
- Many ‘sequential’ defense techniques , if unaware of concurrent programming, are ineffective
- Careful study of Bowspwn and ROPE will yield insights

Findings	Implications
A majority (24 out of 46) of the concurrency attacks corrupt pointer data.	Existing memory safety tools, once made aware of concurrency, may be able to prevent concurrency attacks that corrupt pointer data.
9 concurrency attacks <i>directly</i> corrupt scalar data, such as user identifiers, without compromising memory safety.	Few existing defenses handle attacks that directly corrupt scalar data.
Many existing defenses become unsafe in the face of concurrency errors	These defenses must consider concurrent execution.
The exploitability of a concurrency error highly depends on the duration of its <i>vulnerable window</i> (<i>i.e.</i> , the timing window within which the concurrency error may occur).	New defense techniques may reduce the exploitability of concurrency errors by reducing the duration of the vulnerable window.

Compositional Security

Secure Composition Problem

Composition What can you say about composition of modules A and B?

Parsers, Language Classes & Power

Formal Input Verification Input to parser constitutes valid expression in input-handler's protocol

Secure Composition Prove computational equivalence of input-handling routines, i.e. do two grammars produce exactly the same language? (if not, in extremis birth of 'weird machines')

Requirement Equivalence *undecidable* for complex protocols - starting from language classes that require Non-Deterministic PDA to recognize input language

Way Forward: Minimum Power Principle to Reduce Insecurity of Composition

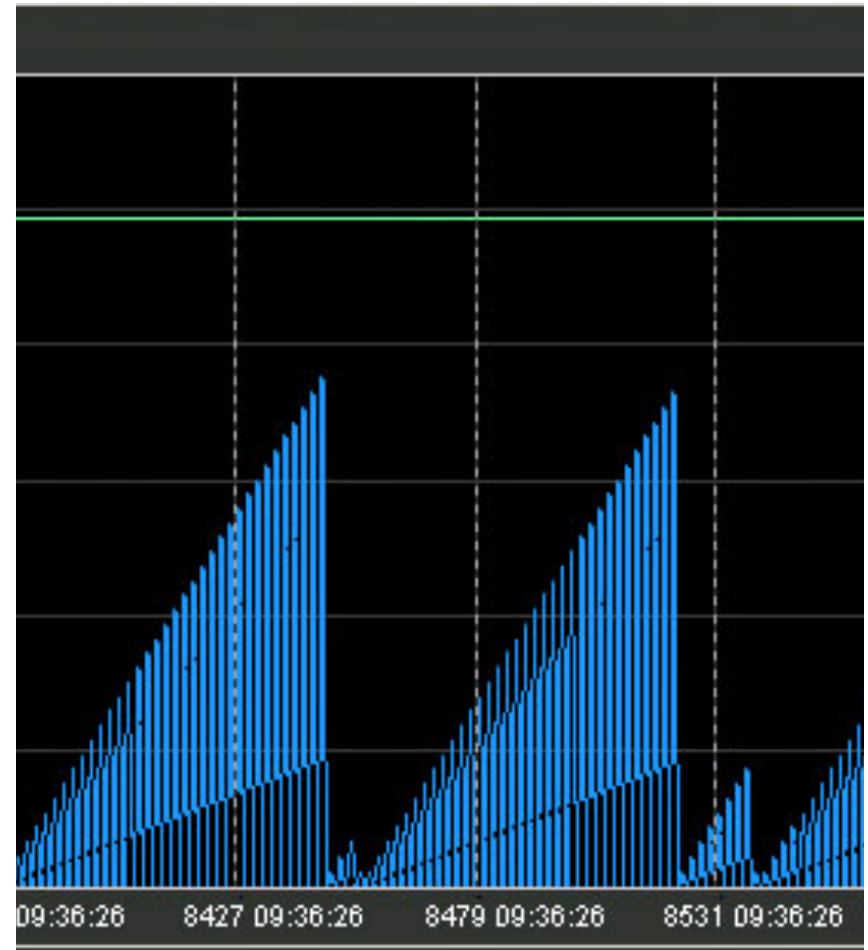
- 1 Parser must *not provide more than the minimal computational strength necessary to interpret the protocol* it is intended to parse
- 2 Protocols should be *designed to require the computationally weakest parser* necessary to achieve the intended operation

DECIDABLE For regular + deterministic context-free grammars

“LangSec” (Nov 2011) IMHO, most fundamental intuition in computer security since Thompson (1984) “Trusting Trust” [LSS11].

Systemic Computer Security

- Motivation: Flash Crash (May 2010) – see my IEEE SP article
- Automated black-box algorithmic trading: Johnson (2013) “Rise of the Machines”
 - phenomenological ‘signatures’ of interacting autonomous computer agents in real-world dynamic (trading) system
 - **All-machine time regime characterized by frequent ‘black swan’ events** with ultrafast durations (<650ms for crashes, <950ms for spikes)
- **Aggregate behavior of simple agents is unpredictable in principle;** no useful security guarantees anent dynamics possible



HFT Nanex 2010

Systemic Computer Security II

- Aggregate behavior of simple agents is unpredictable; no useful security guarantees anent dynamics possible [Joh13] [Bil14]
- Analysis of (side channel) event signatures in phase space; design of circuit breakers, graceful degradation, rectifiers
- Relevance to Singapore 'Smart Cities' (see good/bad example Songdo, Portland)

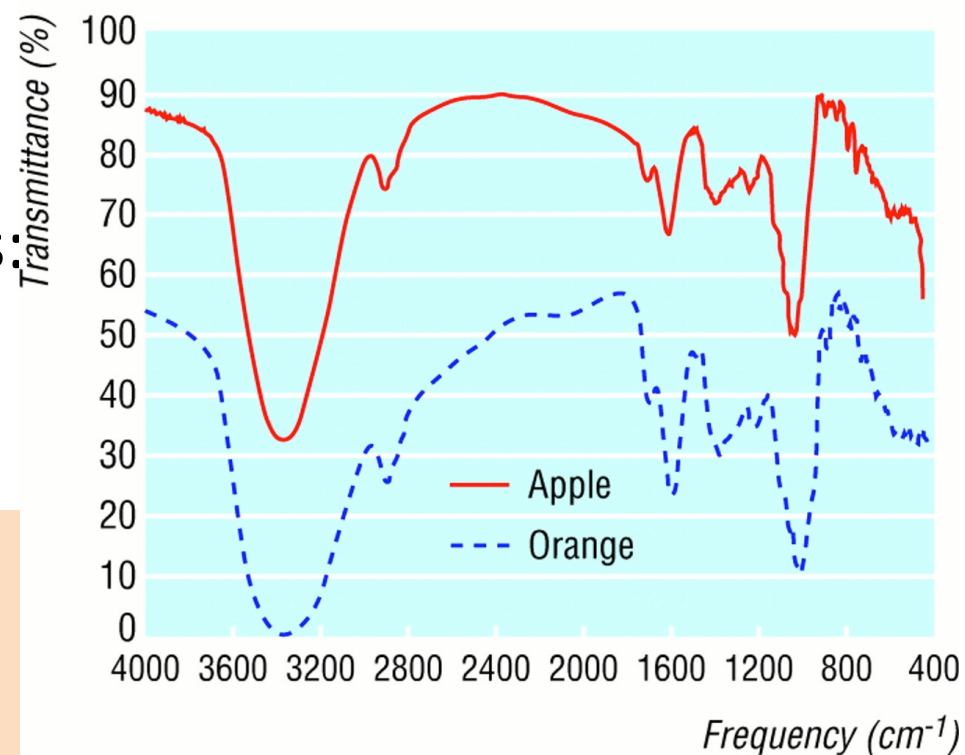
Thank you

How Scientists Relax

- Infrared spectroscopy on a vexing problem of our times: Truly comparing apples and oranges

Thank you for your time and the consideration of ideas.

I appreciate being at SyScan and to finally visit Singapore



A spectrographic analysis of ground, desiccated samples of a Granny Smith apple and a Sunkist navel orange. Picture from [San95]

References I

- [Asan2009] K. Asanovic et al “A view of the parallel computing landscape”, *CACM* 52:10, Oct2009, pp. 56-67 <http://dl.acm.org/citation.cfm?id=1562764.1562783>
- [Boriah2008] S. Boriah et al, “Similarity Measures for Categorical Data”, *SIAM red*, 30:2, 2008 <http://www-users.cs.umn.edu/~sboriah/PDFs/BoriahBCK2008.pdf>
- [Goldw2012] S. Goldwasser and G. Rothblum, "How to Compute in the Presence of Leakage," *FOCS*, Oct. 2012, pp.31-40
<http://eccc.hpi-web.de/report/2012/010/download/>
- [Hund2013] R.Hund et al, “Practical Timing Side Channel Attacks Against Kernel Space ASLR” *IEEE S & P*, 2013 , pp. 191-205
<http://www.ieee-security.org/TC/SP2013/papers/4977a191.pdf>
- [Yang2012] J. Yang et al, “Concurrency attacks”, *USENIX HotPar*, 2012
<https://www.usenix.org/system/files/conference/hotpar12/hotpar12-final44.pdf>
- [JC2013a] M. Jurczyk & G. Coldwind, “Identifying and Exploiting Windows Kernel Race Conditions via Memory Access Patterns”, *SyScan*, April 2013
<http://j00ru.vexillium.org/?p=1695>

References II

- [Dom2012] P. Domingos, "A Few Useful Things to Know about Machine Learning", *CACM* 55:10, Oct 2012, pp. 78-87 <https://t.co/NsAnRUrPtq>
- [Demme2013] J. Demme et al, "On the Feasibility of Online Malware Detection with Performance Counters" *ISCA*, 2013, pp. 559-570 http://www.cs.columbia.edu/~jdd/papers/isca13_malware.pdf
- [Mont2012] D. Montgomery, Design and Analysis of Experiments, Wiley Press, 2012, ch. 1
http://higheredbcs.wiley.com/legacy/college/montgomery/1118146921/supp_material/ch01.doc
- [Murph2012] K. Murphy, Machine Learning, MIT, 2012 <http://www.cs.ubc.ca/~murphyk/MLbook/>
- [Reshef2011] D. Reshef et al. "Detecting Novel Associations in Large Data Sets" *Science* 334.6062 (2011): 1518-1524
<http://www.sciencemag.org/cgi/rapidpdf/334/6062/1518?ijkey=cRCIh2G7AjiA&keytype=ref&siteid=sci>
- [Reshef2011sup] D. Reshef et al. SOM [Reshef2011], *Science* 334.6062 (2011)
<http://www.sciencemag.org/content/334/6062/1502.full?ijkey=l9Qe0i/BE6ZOI&keytype=ref&siteid=sci>
- [Wich2013] G. Wicherski "Taming the ROPE on Sandy Bridge", SyScan, April 2013
<http://www.syscan.org/index.php/download/get/3c6891f2e90e661ea23224cd8f419262/>
[SyScan2013 DAY1 SPEAKER05 Georg Wicherski Taming ROP ON SANDY BRIDGE syscan.zip](#)

References III

- [JC2013b] M. Jurczyk and G. Coldwind, “Kernel double-fetch race condition exploitation on x86 – further thoughts”, *blog*, June 2013, <http://j00ru.vexillum.org/?p=1880>
- [Snyder2008] L. Snyder, “The whole box of tools: William Whewell and the logic of induction”, *Handbook of the History of Logic (British Logic in the Nineteenth Century)*, Ed. : D. Gabbay, Vol 4, 2008, pp.163–228
- [Lipson2009] M. Schmidt and H. Lipson "Distilling Free-Form Natural Laws from Experimental Data," *Science* 324:5923, 2009, pp. 81 – 85
http://ccsl.mae.cornell.edu/sites/default/files/Science09_Schmidt.pdf
- [Gazis2010] P. Gazis et al., “Viewpoints: A High-Performance High-Dimensional Exploratory Data Analysis Tool”, *Publications of the Astronomical Society of the Pacific*, 122(898), 2010, pp. 1518-1525, http://www.giss.nasa.gov/staff/mway/Gazis_Levit_Way2010.pdf
- [PMTK3] K. Murphy, M. Dunham et al, “probabilistic modeling toolkit for Matlab/Octave”, 2011, <https://github.com/probml/pmtk3>
- [San1995] S. Sandford, “Apples and oranges: a comparison”, *Annals of Improbable Research* 1:3, 1995 <http://www.improbable.com/airchives/paperair/volume1/v1i3/air-1-3-apples.html>
- [Alistarh2014] N. Shavit et al, “Are Lock-Free Concurrent Algorithms Practically Wait-Free?”, 2014, <http://research.microsoft.com/pubs/209106/paper.pdf>